ADAPTIVE THRESHOLDING FOR DISTRIBUTED POWER SPECTRUM SENSING

Omar Mehanna, and Nicholas D. Sidiropoulos

Dept. of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN

ABSTRACT

Wideband spectrum sensing is an important prerequisite for cognitive radio access. A network sensing scenario comprising low-end sensors is considered, with each sensor reporting a single *randomly filtered power measurement bit* to the fusion center (FC), which estimates the ambient power spectrum from these bits. An adaptive thresholding algorithm is proposed to improve the quality and speed of power spectrum reconstruction. Upon receipt of each new bit, the FC picks the threshold for the next sensor so as to cut off a half-space from the feasible region along its Chebyshev center. Convergence of this algorithm to the true finite-length autocorrelation is shown, whose Fourier transform yields the power spectrum estimate. To avoid the 'downlink' threshold communication overhead, an alternative algorithm is proposed, where each sensor pseudo-randomly chooses its threshold from a suitable distribution, and the FC judiciously polls sensors to form its power spectrum estimate.

1. INTRODUCTION

Wideband spectrum sensing is a prerequisite for cognitive radio, as it forms the basis for adaptive spectrum sharing. Whereas most work in spectrum sensing has focused on the signal's *Fourier spectrum* (i.e., the Fourier transform of the signal itself), it has recently been argued [1] that estimating the *power spectrum* (the Fourier transform of the signal's autocorrelation) is more relevant for cognitive radio and other applications. Unlike the Fourier spectrum, the power spectrum can be estimated from a finite set of autocorrelation lags using the Fourier transform. This parametrization has been recently explored in [1], where a finite-length autocorrelation vector is estimated by building an over - determined system of linear equations from the (cross-) correlations of the outputs of a bank of periodic modulators, which are linear in the input autocorrelation function.

Here we consider a network sensing scenario, comprising scattered low-end sensors and a fusion center (FC). Each sensor reports a single *randomly filtered power measurement bit* to the FC, which then estimates the ambient power spectrum from the collected bits. One advantage of distributed sensing is that it mitigates the *hidden terminal problem*, shadowing, and fading. Another is that it opens the door for *crowdsourcing* spectrum sensing, using smart phones and other wireless devices.

Power spectrum sensing in this context has been first introduced in [2]. The analog amplitude autocorrelation samples (i.e., very high precision quantization) assumed in [1] cannot be used with scattered low-end sensors that have limited communication capabilities. Nevertheless, it was shown in [2] that adequate power spectrum sensing is still possible from few bits, if one uses a suitable linear programming (LP) formulation that exploits the autocorrelation parametrization and pertinent nonnegativity properties. Note that explicit wideband scanning at high frequency resolution is not possible when only few sensing bits are available; there are not enough bits for all narrow sub-bands, hence more sophisticated aggregate sensing mechanisms must be employed.

In [2], the sensing thresholds at the different sensors were considered to be fixed - in fact a common threshold was chosen for all sensors, which was empirically tuned to ensure that a certain top percentage of sensors generate positive reports. In this paper, we consider adaptive thresholding and polling strategies to improve the quality and speed of power spectrum reconstruction. We first introduce a Chebyshev center adaptive thresholding (CCAT) algorithm, where the threshold for each sensor is iteratively selected so as to cut off a half-space from the presently feasible region along its Chebyshev center, i.e., the center of the largest inscribed ball. By solving an LP to compute each sensing threshold, we show linear convergence of this algorithm to the true finite-length autocorrelation vector, as more sensors report their feedback bits. To avoid the threshold communication overhead, we also introduce a Chebyshev center random thresholding (CCRT) algorithm, where each sensor pseudo-randomly draws its sensing threshold from a suitable distribution. Then, the FC sequentially polls the sensors that can provide the most useful information bits for the estimation problem, while leaving most sensors unpolled. Simulation results illustrate the good estimation performance of CCAT and CCRT, which outperform [2].

Our problem formulation may be reminiscent of *one-bit compressed sensing* [3, 4]. It has been shown in [3] that signals can be recovered with good accuracy from compressive sensing measurements quantized to just one bit per measurement. The flip-side is that the number of one-bit measurements required for accurate reconstruction appears to be rather large, and the algorithm is more cumbersome than in the case of real-valued measurements. This one-bit framework has been extended in [4], where the signal is recovered using a generalized approximate message passing (GAMP) algorithm, and the quantization threshold is selected for each measurement such that it partitions the consistent region along its centroid. The difference of our setup with one-bit compressed sensing is that the latter does not exploit additional autocorrelation-specific constraints, which are important in our context, and it requires signal sparsity, which is not necessary in our setup. In addition, the GAMP algorithm assumes a signal with a separable distribution that is known a-priori, which is not a valid assumption in general.

2. POWER SPECTRUM SENSING FROM FEW BITS

Wideband power spectrum sensing from few bits was first introduced in [2], where the term *Frugal Sensing* was coined to underscore that the approach targets heavily under-determined wideband sensing scenarios involving only low-end sensors and low-rate communication. Consider M scattered sensors measuring the ambient signal power and reporting to a FC. It is assumed that every sensor processes the same wide-sense stationary (WSS) received signal, $x(t)$, which is sampled using an analog-to-digital converter operating at Nyquist rate, yielding the discrete-time signal $x(n)$. Differ-

Contact: (meha0006,nikos)@umn.edu. Supported in part by NSF CCF 0747332, ECCS 1231504.

ences in path loss can be factored out via automatic gain control, and frequency-selective fading can be mitigated by averaging the measurements over longer periods, as shown in [2]. The Nyquist sampling requirement can be lifted by using an equivalent analog processing and integration chain [2]. Note that the power spectrum is invariant with respect to timing offsets and phase shifts.

Sensor $m \in \{1, \ldots, M\}$ then passes $x(n)$ through a wideband finite impulse response (FIR) filter with random complex pseudo-noise (PN) impulse response $g_m(n)$ of length K:

$$g_m(n) = \begin{cases} (1/\sqrt{2K})(\pm 1 \pm j) & \text{if} \quad 0 \le n \le K-1 \\ 0 & \text{otherwise} \end{cases}$$

The filter sequence $g_m(n)$ can be generated using a PN linear shift register, whose initial seed is unique for each sensor (e.g., its serial number), and is assumed to be known to the FC. Using random PN filters can also be motivated from a random projections viewpoint, as for the compression matrix applied to sparse signals [3]. The filter's output sequence is expressed as $z_m(n) = \sum_{k=0}^{K-1} g_m(k)x(n-k)$. Let $\alpha_m := \mathbb{E}[|z_m(n)|^2]$ denote the average power of the WSS signal $z_m(n)$. Each sensor estimates α_m using a sample average:

$$\hat{\alpha}_m = \frac{1}{N} \sum_{n=0}^{N-1} |z_m(n)|^2 \tag{1}$$

with $\lim_{N\to\infty} \hat{\alpha}_m = \alpha_m$ under appropriate mixing conditions. Finally, each sensor compares the estimated $\hat{\alpha}_m$ to a sensor-specific threshold t_m. If $\hat{\alpha}_m \ge t_m$, then sensor m sends a bit $b_m = 1$ to the FC, otherwise it sends $b_m = 0$.

Let $r_x(\ell) := \mathbb{E}[x(n)x^*(n-\ell)]$ denote the autocorrelation sequence of $x(n)$, define

$$q_m(\ell) := \sum_{n=0}^{K-\ell-1} g_m(n+\ell)g_m^*(n), \quad \ell = 0, \ldots, K-1,$$

and define the following vectors:

$$\begin{aligned}
\mathbf{g}_m &:= [g_m^*(K-1), g_m^*(K-2), \ldots, g_m^*(0)]^T \\
\mathbf{x}_n &:= [x(n), x(n+1), \ldots, x(n+K-1)]^T \\
\mathbf{r}_x &:= [r_x(1-K), \ldots, r_x(-1), r_x(0), r_x(1), \ldots, r_x(K-1)]^T \\
\tilde{\mathbf{r}}_x &:= [r_x(0), \text{Re}\{r_x(1)\}, \ldots, \text{Re}\{r_x(K-1)\}, \text{Im}\{r_x(1)\}, \\
&\quad \ldots, \text{Im}\{r_x(K-1)\}]^T \\
\mathbf{q}_m &:= [q_m(0), 2\,\text{Re}\{q_m(1)\}, \ldots, 2\,\text{Re}\{q_m(K-1)\}, \\
&\quad 2\,\text{Im}\{q_m(1)\}, \ldots, 2\,\text{Im}\{q_m(K-1)\}]^T
\end{aligned}$$

where $\text{Re}\{\cdot\}$ and $\text{Im}\{\cdot\}$ denote the real and imaginary parts, respectively. It can be shown that

$$\begin{aligned}
\alpha_m &= \mathbb{E}[|z_m(n)|^2] = \mathbb{E}[|\mathbf{x}_n^H \mathbf{g}_m|^2] \\
&= \mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m = \mathbf{q}_m^T \tilde{\mathbf{r}}_x
\end{aligned} \tag{2}$$

where $\mathbf{R}_x$ is the $K \times K$ Toeplitz autocorrelation matrix. Thus, upon receipt of $b_m = 1$ (or $b_m = 0$) from sensor m, the FC learns that $\mathbf{q}_m^T \tilde{\mathbf{r}}_x \ge t_m$ (resp. $\mathbf{q}_m^T \tilde{\mathbf{r}}_x < t_m$), assuming sufficient averaging such that sample averages converge to ensemble averages. Note that since we only need to ensure that the inequality is not reversed, sample averaging requirements are considerably relaxed relative to high-rate quantization.

A windowed estimate of the power spectrum can be obtained from the K-lag autocorrelation as $\hat{S}_x(\omega) = \sum_{\ell=-K+1}^{K-1} r_x(\ell)e^{-j\omega\ell}$. A discrete N_F-point estimate of the power spectrum can be obtained as $\hat{\mathbf{s}}_x = \mathbf{F}\mathbf{r}_x$, where $\hat{s}_x(f) = \hat{S}_x\left(\frac{2\pi f}{N_F}\right)$, $f = 0, \ldots, N_F - 1$, and $\mathbf{F}$ is the $N_F \times (2K-1)$ (phase-shifted) discrete Fourier transform (DFT) matrix. Defining

$$\mathbf{Q} := \begin{bmatrix} \mathbf{0}_{K-1} & \tilde{\mathbf{I}}_{K-1} & -j\,\tilde{\mathbf{I}}_{K-1} \\ 1 & \mathbf{0}_{K-1}^T & \mathbf{0}_{K-1}^T \\ \mathbf{0}_{K-1} & \mathbf{I}_{K-1} & j\,\mathbf{I}_{K-1} \end{bmatrix} \tag{3}$$

where $\mathbf{0}_{K-1}$ is a $(K-1)$ zeros vector, $\mathbf{I}_{K-1}$ is the $(K-1)$ identity matrix, and $\tilde{\mathbf{I}}_{K-1}$ is the *flipped* $(K-1)$ identity matrix, then it easy to see that $\mathbf{r}_x = \mathbf{Q}\tilde{\mathbf{r}}_x$. Thus the discrete power spectrum estimate can be obtained from $\tilde{\mathbf{r}}_x$ as $\hat{\mathbf{s}}_x = \mathbf{F}\mathbf{Q}\tilde{\mathbf{r}}_x = \tilde{\mathbf{F}}\tilde{\mathbf{r}}_x$, where $\tilde{\mathbf{F}} := \mathbf{F}\mathbf{Q}$. *Therefore, the goal is to estimate the real vector $\tilde{\mathbf{r}}_x$ at the FC from the few measurement bits $\{b_m\}$ received from the M sensors.*

An LP formulation that minimizes the total signal power and encourages sparsity has been proposed in [2] to estimate $\tilde{\mathbf{r}}_x$, using equal sensing thresholds for all sensors. In the following section, we propose an adaptive thresholding technique, and show that it can significantly improve the power spectrum estimate.

3. ADAPTIVE THRESHOLDING

Consider a time-slotted bi-directional communication link between the M sensors and the FC, comprising M time slots. At the beginning of each time slot $m \in \{1, \ldots, M\}$, the FC sends the threshold t_m to sensor m. Sensor m then compares it's measured α_m with t_m, and responds with either $b_m = 1$ or $b_m = 0$ within the same slot.

Prior to receiving any information bits from the sensors, properties of the autocorrelation can be exploited to define an initial feasible region for $\tilde{\mathbf{r}}_x$. First, an upper bound $P_{\max}$ for the total signal power can be easily obtained from historical data yielding the bounds: $0 \le r_x(0) \le P_{\max}$. Another well-known property is that $|r_x(\ell)| \le r_x(0)$, for $\ell = 1, \ldots, K-1$. These inequalities define a *bounded* polyhedron as the initial feasible region for $\tilde{\mathbf{r}}_x$:

$$\begin{aligned}
\mathcal{P}_0 := \big\{\mathbf{y} \in \mathbb{R}^{(2K-1)} \,|\, 0 \le y(0) \le P_{\max}, -y(0) \le y(\ell) \le y(0), \\
\ell = 1, \ldots, 2K-2\big\}.
\end{aligned} \tag{4}$$

When the FC receives all measurement bits $\{b_m\}_{m=1}^M$, the final feasible region is defined by the polyhedron (ignoring α_m estimation errors):

$$\begin{aligned}
\mathcal{P}_M \;=\; \mathcal{P}_0 \cap \big\{\mathbf{y} \,|\, \mathbf{q}_m^T\mathbf{y} \ge t_m \text{ if } b_m = 1, \\
\mathbf{q}_m^T\mathbf{y} < t_m \text{ if } b_m = 0,\ m = 1, \ldots, M\big\}.
\end{aligned} \tag{5}$$

The volume of the feasible region $\mathcal{P}_M$ gives a measure of ignorance or uncertainty about $\tilde{\mathbf{r}}_x \in \mathcal{P}_M$; a small $\mathcal{P}_M$ implies that $\tilde{\mathbf{r}}_x$ is localized to within a small set, whereas a large $\mathcal{P}_M$ means that there is still much uncertainty about $\tilde{\mathbf{r}}_x$. In other words, a smaller feasible region $\mathcal{P}_M$ translates to higher accuracy in localizing $\tilde{\mathbf{r}}_x$. *Thus, our objective is to adaptively select the thresholds $\{t_m\}_{m=1}^M$ to ensure that $\mathcal{P}_M$ is as small as possible.*

The proposed Chebyshev center adaptive thresholding (CCAT) algorithm, which is implemented at the FC, can be explained as follows:

Given the initial polyhedron $\mathcal{P}_0$, its Chebyshev center $\mathbf{y}_{cc}^{(0)}$, and $\{\mathbf{q}_m\}_{m=1}^M$. *For* each time-slot/sensor $m = 1, \ldots, M$, *do*

1. Set the threshold $t_m = \mathbf{q}_m^T \mathbf{y}_{cc}^{(m-1)}$, and send it to sensor m requesting its measurement bit b_m.
2. Upon receiving b_m, update the feasible polyhedron:
$$\mathcal{P}_m := \begin{cases} \mathcal{P}_{m-1} \cap \{\mathbf{y} \,|\, \mathbf{q}_m^T\mathbf{y} \ge t_m\} & \text{if } b_m = 1 \\ \mathcal{P}_{m-1} \cap \{\mathbf{y} \,|\, \mathbf{q}_m^T\mathbf{y} < t_m\} & \text{if } b_m = 0 \end{cases} \tag{6}$$
3. Find the Chebyshev center, $\mathbf{y}_{cc}^{(m)}$, of $\mathcal{P}_m$.

Finally, an estimate $\hat{\mathbf{r}}_x$ of $\tilde{\mathbf{r}}_x$ can be directly obtained as the Chebyshev center of $\mathcal{P}_M$, i.e., $\hat{\mathbf{r}}_x = \mathbf{y}_{cc}^{(M)}$.

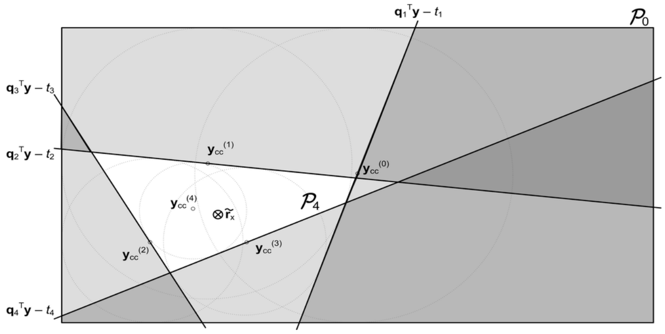

Fig. 1. Illustrative example for the CCAT algorithm.

The Chebyshev center of a polyhedron defined by a set of L linear inequalities, $\mathcal{P} := \{\mathbf{y} \mid \mathbf{a}_i^T\mathbf{y} \le d_i,\ i = 1, \ldots, L\}$, can be found by solving the LP [7, Sec. 8.5.1]:

$$\boxed{\begin{array}{ll} \max\limits_{R \ge 0, \mathbf{y}} & R \\ \text{s.t.}: & \mathbf{a}_i^T\mathbf{y} + R||\mathbf{a}_i||_2 \le d_i,\ i = 1, \ldots, L \end{array}} \qquad (7)$$

The Chebyshev center is the point inside $\mathcal{P}$ that has the maximum distance to the closest point in the boundary hyperplanes defining $\mathcal{P}$ (i.e., the exterior of $\mathcal{P}$), and it is also the center of the largest ball that lies inside $\mathcal{P}$ [7].

Note that at the second step of the CCAT algorithm, the half-space $\{\mathbf{y} \mid \mathbf{q}_m^T(\mathbf{y} - \mathbf{y}_{cc}^{(m-1)}) < 0\}$ (or $\{\mathbf{y} \mid \mathbf{q}_m^T(\mathbf{y} - \mathbf{y}_{cc}^{(m-1)}) \ge 0\}$) is cut-off from the feasible region if $b_m = 1$ (resp. $b_m = 0$). The selection of the threshold $t_m = \mathbf{q}_m^T\mathbf{y}_{cc}^{(m-1)}$ ensures that the Chebyshev center of $\mathcal{P}_{m-1}$ is a point in the trimmed half-space. This ensures that a large portion of $\mathcal{P}_{m-1}$ is omitted from the feasible region, and that the new polyhedron $\mathcal{P}_m$ is considerably smaller than $\mathcal{P}_{m-1}$. An illustrative example for the CCAT algorithm in $\mathbb{R}^2$ is given in Fig. 1, with $M = 4$, and a rectangular $\mathcal{P}_0$. The grey shaded regions in the figure represent the union of planes inside $\mathcal{P}_0$ that are cut-off from the feasible region after each b_m is received, whereas the final (small) feasible region $\mathcal{P}_4$, which includes $\tilde{\mathbf{r}}_x$, is unshaded. It is worth noting that a similar approach has been used in solving general convex and quasiconvex optimization problems [5, 6].

It can be shown that $\mathbf{y}_{cc}^{(M)} \to \tilde{\mathbf{r}}_x$ as $M \to \infty$ using the proposed CCAT algorithm, for $\{\mathbf{q}_m\}_{m=1}^M$ drawn from a distribution that satisfies certain conditions, and that the algorithm has a linear rate of convergence. The convergence follows from Theorem 1 in [5], which states that the sequence $\{\rho_m\}_{m=1}^\infty$, where ρ_m is the radius of the largest ball that lies inside $\mathcal{P}_m$, converges to zero. The complete convergence proof will be presented in the journal version of this work. Note that plotting the error $||\mathbf{y}_{cc}^{(m)} - \tilde{\mathbf{r}}_x||^2$ as a function of m shows the number of bits/senors required to obtained an estimate $\hat{\mathbf{r}}_x$ that is within a certain ϵ estimation error.

The number of linear inequalities defining $\mathcal{P}_m$ increases at each iteration of the algorithm, and hence the computational effort to compute $\mathbf{y}_{cc}^{(m)}$ increases. For a polyhedron that is defined by L linear inequalities, one approach is to keep only a fixed number $J \le L$ of the most relevant inequality constraints while dropping the other $L - J$ less relevant or redundant inequalities [5, 6]. We use the heuristic method described in [6] to rank the L constraints, such that only the J top-ranked inequalities are used in computing the Chebyshev center of the polyhedron at each iteration. With proper choice of J ($> 10K$), simulations have shown a negligible effect on the performance, at a dramatic decrease in the total computation time of the CCAT algorithm.

Instead of using the Chebyshev center of the polyhedron $\mathcal{P}_{m-1}$ in computing t_m, other options include the center of gravity, the center of the maximum volume inscribed ellipsoid, and the analytic center [6]. The complexity of each method can be captured by the worst-case number of iterations (i.e., M) required to achieve an ϵ-error estimate, $||\hat{\mathbf{r}}_x - \tilde{\mathbf{r}}_x||^2 \le \epsilon$, in addition to the complexity of computing each center at each iteration. Using the center of gravity, the volume of the polyhedron is guaranteed to reduce by at least 37% at each iteration of the algorithm, and the number of iterations required to achieve an ϵ-error estimate is at most $O\left(K \log(1/\epsilon)\right)$. However, computing the center of gravity of a polyhedron described by a set of linear inequalities is NP-hard [6]. The number of iterations required using the center of the maximum volume inscribed ellipsoid is at most $O\left(K^2 \log(1/\epsilon)\right)$ [6]. Although the center of the maximum volume inscribed ellipsoid is computed by solving a convex problem, its computation is prohibitive for large K [7]. The analytic center can be efficiently computed [7], but at most $O\left(K^2/\epsilon^2\right)$ iterations are required to achieve an ϵ-error [8]. Whereas no similar convergence analysis has been developed for the Chebyshev center (probably because the Chebyshev center can be strongly affected by scaling or affine transformations of coordinates), exhaustive simulations showed that the performance of CCAT algorithm is almost identical to the case of using the analytic center, with much smaller computation time to solve the LP (7).

If the number of sensors/time-slots M is small, and each sensor is limited to send only a single bit, then it is recommended to include the linear positivity constraint $\tilde{\mathbf{F}}\mathbf{y} \ge \mathbf{0}$ to the initial polyhedron $\mathcal{P}_0$. Including this constraint results in a smaller-volume $\mathcal{P}_0$, thus achieving better estimates $\hat{\mathbf{r}}_x$ for small M. However, the convergence $\mathbf{y}_{cc}^{(M)} \to \tilde{\mathbf{r}}_x$ for large M is no longer ensured, since $\tilde{\mathbf{F}}\tilde{\mathbf{r}}_x$ can have negative values when the autocorrelation is truncated to finite K-lags. The pros and cons of including $\tilde{\mathbf{F}}\mathbf{y} \ge \mathbf{0}$ in $\mathcal{P}_0$ are further illustrated in the simulations in Section 5.

4. RANDOM THRESHOLDS AND SENSOR POLLING

The proposed CCAT algorithm requires that the FC sends a computed threshold to each sensor at the beginning of its allocated time-slot. To avoid this threshold communication overhead, it may be more appealing to pre-assign the thresholds to the sensors.

We propose that each sensor randomly selects t_m from a Gaussian distribution with the same mean and variance as α_m, which is a random variable with respect to the random signal and the PN filter of each sensor. This choice implies that the volume of $\mathcal{P}_M$ is smaller, on average, for $\tilde{\mathbf{r}}_x$ that is closer to its expectation. Localizing $\tilde{\mathbf{r}}_x$ inside a small-volume polyhedron guarantees that the estimate $\hat{\mathbf{r}}_x$ is close to $\tilde{\mathbf{r}}_x$. This is not the case, for example, if the thresholds are chosen from a uniform distribution, or are fixed. The mean and variance of α_m can be obtained from historical data across different signals and filters. The FC is assumed to know $\{t_m\}$, because they are pseudo-randomly generated at the sensors, e.g., based on sensor ID as the seed for the PN generator.

Now, the FC sequentially selects the important sensors to poll, requesting their feedback bit. Consider a time-slotted structure similar to the CCAT algorithm, where at the beginning of each time-slot the FC polls one sensor, which in turn responds with its feedback bit within the same time slot. Let $\mathcal{J}$ denote the set of sensors that have sent their feedback bits while $\bar{\mathcal{J}}$ denotes the set of remaining sensors. The proposed Chebyshev center random thresholding (CCRT) algorithm with pre-assigned thresholds can be described as follows:

Given the sets $\{\mathbf{q}_m\}_{m=1}^M$, $\{t_m\}_{m=1}^M$, the initial polyhedron $\mathcal{P}_0$, and its Chebyshev center $\mathbf{y}_{cc}^{(0)}$. Initialize $k = 1$. *While* $k \le M$*, do*

1. For each $m \in \bar{\mathcal{J}}$, find the distance between the hyperplane $\{\mathbf{y} \mid \mathbf{q}_m^T \mathbf{y} - t_m = 0\}$ and the Chebyshev center $\mathbf{y}_{cc}^{(k-1)}$:
$$d_m = \frac{|\mathbf{q}_m^T \mathbf{y}_{cc}^{(k-1)} - t_m|}{||\mathbf{q}_m||} \quad (8)$$
2. Select the senor $m^* = \arg\min d_m$ to be polled requesting its measurement bit b_{m^*}.
3. Upon receiving b_{m^*}, delete m^* from $\bar{\mathcal{J}}$, add it to $\mathcal{J}$, and update the polyhedron:
$$\mathcal{P}_k := \begin{cases} \mathcal{P}_{k-1} \cap \{\mathbf{y} \mid \mathbf{q}_{m^*}^T \mathbf{y} \geq t_{m^*}\} & \text{if } b_{m^*} = 1 \\ \mathcal{P}_{k-1} \cap \{\mathbf{y} \mid \mathbf{q}_{m^*}^T \mathbf{y} < t_{m^*}\} & \text{if } b_{m^*} = 0 \end{cases} \quad (9)$$
4. Find the Chebyshev center $\mathbf{y}_{cc}^{(k)}$ of $\mathcal{P}_k$ by solving the LP (7).
5. *Terminate* if $\mathbf{y}_{cc}^{(k)} = \mathbf{y}_{cc}^{(k-1)}$ for the last β consecutive iterations (β is a design parameter). Else, increment k and repeat.

Finally, an estimate of $\tilde{\mathbf{r}}_x$ can be directly obtained as $\hat{\mathbf{r}}_x = \mathbf{y}_{cc}^{(\bar{k})}$, where $\bar{k}$ is the last k at the algorithm's termination.

Note that by polling sensor m^* with the smallest distance between $\mathbf{y}_{cc}^{(k-1)}$ and the hyperplane $\{\mathbf{y} \mid \mathbf{q}_m^T \mathbf{y} - t_m = 0\}$ at the k-th iteration, we try to ensure that the Chebyshev center of $\mathcal{P}_{k-1}$ is very close to the hyperplane which defines the trimmed half-space. The Chebyshev center $\mathbf{y}_{cc}^{(k-1)}$ can be inside or outside the cut-off half-space. Similar to the CCAT algorithm, this ensures that a large portion of $\mathcal{P}_{k-1}$ is omitted from the feasible region, and that the updated polyhedron $\mathcal{P}_k$ is considerably smaller than $\mathcal{P}_{k-1}$.

For sufficiently large M, the performance of the CCRT algorithm is similar to the performance of the CCAT algorithm since $d_{m^*} \to 0$ as $M \to \infty$. In other words, if M is sufficiently large, then at each iteration k there exists a sensor $m \in \bar{\mathcal{J}}$ with $d_m \approx 0$, and thus the CCRT algorithm becomes almost identical to the CCAT algorithm. For small M, however, as more sensors are polled, it becomes harder to find sensors $m \in \bar{\mathcal{J}}$ with small d_m. After some k iterations, d_{m^*} becomes relatively large such that the half-space information obtained by polling any of the remaining sensors $m \in \bar{J}$ is redundant, and thus the feasible region polyhedron will not decrease (i.e., $\mathcal{P}_k = \mathcal{P}_{k-1}$). When this limit is reached, the Chebyshev center does not change, and thus the CCRT algorithm can be terminated prematurely. Note that we need to check that $\mathbf{y}_{cc}^{(k)} = \mathbf{y}_{cc}^{(k-1)}$ for the last $\beta \geq 1$ iterations, since for some scenarios polling senor $m^* = \arg\min d_m$ does not change the Chabyshev center, whereas polling another sensor $\bar{m} \in \bar{\mathcal{J}} - \{m^*\}$, where $d_{\bar{m}} > d_{m^*}$, yields a smaller polyhedron with a different Chebyshev center. A small $\beta = 5 \sim 10$ is apparently sufficient. Due to the larger feasible region obtained at the termination of the CCRT algorithm with limited M, the estimation error with the CCRT algorithm is generally larger than that obtained with the CCAT algorithm.

5. NUMERICAL RESULTS

A filter length $K = 12$ and $M = 300$ sensors were considered for the default CCAT and CCRT algorithm setups, and a sample-size $N = 6 \times 10^4$ is used by each sensor to compute α_m. To measure the quality of the estimate, we use the normalized mean square error (NMSE), defined as $\text{NMSE}(k) := \mathbb{E}\left[\frac{||\tilde{\mathbf{r}}_x - \hat{\mathbf{r}}_x^{(k)}||^2}{||\tilde{\mathbf{r}}_x||^2}\right]$, where $\hat{\mathbf{r}}_x^{(k)}$ is the estimate of $\tilde{\mathbf{r}}_x$ when the FC receives the k-th bit, $k = 1, \ldots, M$. The expectation is taken with respect to the random signals and the random impulse responses of the FIR filters, obtained via more than 200 Monte-Carlo simulations. The NMSE as a function of the number of received bits k is plotted in Fig. 2. Fig. 2 shows that $\text{NMSE}(k) \to 0$ (i.e., $\hat{\mathbf{r}}_x^{(k)} \to \tilde{\mathbf{r}}_x$) as k increases with the CCAT algorithm, and confirms the linear rate of convergence. The figure also shows that the NMSE obtained when using the analytic center instead of the Chebyshev center (i.e., ACAT algorithm), plotted as the dotted line, is very similar to (slightly worse than) the NMSE of the CCAT algorithm, while computation time for the CCAT algorithm is much lower. When the positivity constraint $\tilde{\mathbf{F}}\hat{\mathbf{r}}_x \geq \mathbf{0}$ was included to the initial polyhedron $\mathcal{P}_0$, the NMSE of the CCAT was better than the default setting without the positivity constraint for $M < 150$. But as more bits are received, the NMSE of the CCAT with the positivity constraint saturates at 0.007, whereas the NMSE of the CCAT without the positivity constraint continues decreasing to zero. This shows that including the positivity constraint improves the performance only up to moderate M; for higher M it should be omitted to enable convergence. When the filter length was increased to $K = 16$, more sensors were needed to achieve the same NMSE level obtained with $K = 12$, due to the additional number of unknowns that need to be estimated. Note that, on the other hand, the resolution of the estimated power spectrum increases with K.

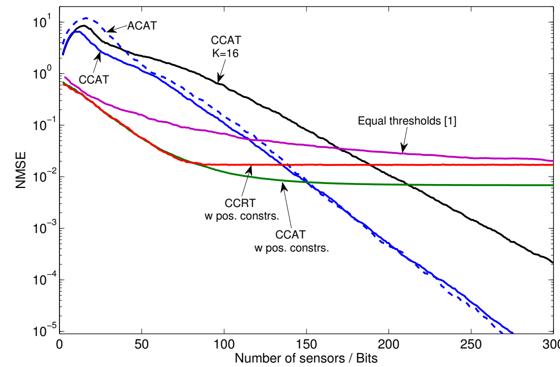

Fig. 2. Performance of the CCAT and CCRT algorithms.

For the CCRT algorithm, each t_m was randomly drawn from a Gaussian with the same mean and variance as the random α_m, obtained via Monte-Carlo simulations. Fig. 2 shows that, for the first 80 polled sensors, the NMSE of the CCRT algorithm is similar to the NMSE of the CCAT algorithm; thereafter the NMSE of the CCRT algorithm saturates at 0.017. The NMSE obtained using equal thresholds (with 100 sensors reporting $b_m = 1$), and applying the $\tilde{\mathbf{r}}_x$-estimation method proposed in [2], is also plotted in Fig. 2, yielding a larger NMSE compared to the CCAT and CCRT algorithms. It is worth mentioning that almost identical results were obtained when only the most relevant $J = 150$ inequality constraints were considered when solving (7), using the constraint dropping technique mentioned in Section 3.

6. CONCLUSIONS

We have introduced an adaptive thresholding algorithm that enables the FC to estimate the ambient power spectrum from bits it receives from scattered low-end sensors. By adapting the threshold for each sensor so as to cut off a half-space from the feasible region along its Chebyshev center, we derived an algorithm that outperforms [2], which uses static threshold selection. An alternative pseudo-random thresholding and polling algorithm was proposed to avoid the threshold communication overhead. Our results underscore the importance of judicious threshold design / adaptation in the context of distributed power spectrum sensing.

7. REFERENCES

[1] D.D. Ariananda, and G. Leus, "Compressive Wideband Power Spectrum Estimation," *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4775–4789 , Sep. 2012.

[2] O. Mehanna and N. D. Sidiropoulos, "Frugal sensing: wideband power spectrum sensing from few bits," *IEEE Transactions on Signal Processing,* June 2013 (To appear).

[3] P. Boufounos and R. Baraniuk, "1-bit compressive sensing," *in Proc. Conf. Inform. Science and Systems (CISS),* Princeton, NJ, Mar. 2008.

[4] U. Kamilov, A. Bourquard, A. Amini, and M. Unser, "One-bit measurements with adaptive thresholds" *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 607–610, Oct. 2012.

[5] J. Elzinga and T. G. Moore, "A central cutting plane algorithm for the convex programming problem," *Mathematical Programming Studies,* vol. 8, pp. 134–145, 1975.

[6] S. Boyd and L. Vandenberghe "Localization and cutting-plane methods," EE364b Lecture Notes, Stanford University, 2007.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[8] Yinyu Ye, "Complexity analysis of the analytic center cutting plane method that uses multiple cuts," *Mathematical Programming,* vol. 78, pp. 85–104, 1997.